NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SCALE-INDEPENDENT CLUSTERING METHOD WITH AUTOMATIC VARIABLE SELECTION BASED ON TREES

by

Sarah K. Lynch

March 2014

Thesis Advisor:	Samuel E. Buttrey
Second Reader:	Lyn R. Whitaker

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2014	**3. REPORT TYPE AND DATES COVERED** Master’s Thesis

4. TITLE AND SUBTITLE A SCALE-INDEPENDENT CLUSTERING METHOD WITH AUTOMATIC VARIABLE SELECTION BASED ON TREES	**5. FUNDING NUMBERS**
6. AUTHOR(S) Sarah K. Lynch	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Clustering is the process of putting observations into groups based on their distance, or dissimilarity, from one another. Measuring distance for continuous variables often requires scaling or monotonic transformation. Determining dissimilarity when observations have both continuous and categorical measurements can be difficult because each type of measurement must be approached differently.

We introduce a new clustering method that uses one of three new distance metrics. In a dataset with p variables, we create p trees, one with each variable as the response. Distance is measured by determining on which leaf an observation falls in each tree. Two observations are similar if they tend to fall on the same leaf and dissimilar if they are usually on different leaves.

The distance metrics are not affected by scaling or transformations of the variables and easily determine distances in datasets with both continuous and categorical variables. This method is tested on several well-known datasets, both with and without added noise variables, and performs very well in the presence of noise due in part to automatic variable selection. The new distance metrics outperform several existing clustering methods in a large number of scenarios.

14. SUBJECT TERMS: Clustering, Regression Trees, Classification Trees, Cramér’s *V*	**15. NUMBER OF PAGES** 49
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

A SCALE-INDEPENDENT CLUSTERING METHOD WITH AUTOMATIC VARIABLE SELECTION BASED ON TREES

Sarah K. Lynch
Lieutenant, United States Navy
B.A., University of Rochester, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: Sarah K. Lynch

Approved by: Samuel E. Buttrey
Thesis Advisor

Lyn R. Whitaker
Second Reader

Robert F. Dell
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Clustering is the process of putting observations into groups based on their distance, or dissimilarity, from one another. Measuring distance for continuous variables often requires scaling or monotonic transformation. Determining dissimilarity when observations have both continuous and categorical measurements can be difficult because each type of measurement must be approached differently.

We introduce a new clustering method that uses one of three new distance metrics. In a dataset with *p* variables, we create *p* trees, one with each variable as the response. Distance is measured by determining on which leaf an observation falls in each tree. Two observations are similar if they tend to fall on the same leaf and dissimilar if they are usually on different leaves.

The distance metrics are not affected by scaling or transformations of the variables and easily determine distances in datasets with both continuous and categorical variables. This method is tested on several well-known datasets, both with and without added noise variables, and performs very well in the presence of noise due in part to automatic variable selection. The new distance metrics outperform several existing clustering methods in a large number of scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AGNES	agglomerative nesting
Bagging	bootstrap aggregation
c&rt	classification and regression trees
DIANA	divisive analysis
PAM	partitioning around medoids
One-SE	one standard error

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Clustering is the process of grouping observations with other observations which share similar characteristics (Hartigan, 1975, p. 1). Observations in a cluster should be similar to one another and different from observations in other clusters (Mirkin, 2005, pp. ix–x). A way to measure dissimilarity between observations is by measuring their distance from one another. Current clustering algorithms usually use the Euclidean or Manhattan distances as a measurement of dissimilarity (Kaufman & Rousseeuw, 1990, p. 11–12).

There are several problems with the current distance metrics used for clustering. Continuous variables often require scaling or monotonic transformation (Kaufmann & Rousseeuw, 1990, p. 4). Categorical variables have to undergo their own transformation into a useable form from which distances can be calculated. Clustering datasets with mixed data, both continuous and categorical variables, can be difficult because each measurement must be treated differently (Mirkin, 2005, pp. 65–66). We introduce a new clustering method that easily clusters mixed data, is not affected by scaling or transformation, and performs automatic variable selection, allowing it to perform well in the presence of noise.

Our clustering method uses one of three new distance metrics, d_1, d_2, and d_3, calculated by using classification and regression trees. In a dataset with n observations and p variables, we build p trees, one with each variable in the dataset as the response. Distance between observations is measured by determining in which leaves observations fall with respect to one another. Two observations are similar if they are in the same leaf and are different if they are in different leaves. After our method calculates all of the distances between observations, we run the agglomerative nesting (AGNES) clustering algorithm on the distances to determine the clusters (Kaufman & Rousseeuw, 1990, p. 199).

We tested our new method on three well-known datasets from the University of California at Irvine Machine Repository. The true number of classes for each dataset is already known, allowing us to test our method’s predictive ability (Bache & Lichman, 2014). We added 15 and 50 variables of random noise to each dataset to see how well our

method performs in the presence of noise. Although we know the number of classes for each dataset, more clusters might actually occur in nature; therefore, we clustered a dataset with *k* classes using *k* and 2*k* clusters.

To test how well our new method is able to accurately cluster data, we compared all three distance metrics to four other clustering algorithms: Partitioning around medoids (PAM), divisive analysis (DIANA), AGNES (Kaufmann & Rousseeuw, 1990), and the *K*-means partitioning algorithm (Hartigan, 1975). We then calculated Cramér's *V* to evaluate the quality of each clustering solution.

At least one of the new distance metrics performed better than the other four algorithms in over 77 percent of the 18 cases. AGNES performed equally as well on the Iris dataset with no noise and three clusters. The *K*-means algorithm performed better than the new distance metrics on the Optical dataset in every case when using 20 clusters. In cases when one of the new distance metrics outperformed the other algorithms, d_2 did the best 66 percent of the time.

Our method performs very well in the presence of noise compared to the other clustering algorithms. The use of classification and regression trees on all *p* variables removes any need to scale or transform any of the data and allowed automatic variable selection.

LIST OF REFERENCES

Bache, K. & Lichman, M. (2014). *UCI Machine Learning Repository*. Retrieved from University of California at Irvine, School of Information and Computer Science website: http://archive.ics.uci.edu/ml/datasets.html

Hartigan, J. (1975). *Clustering algorithms*. New York: John Wiley and Sons

Kaufman, L.& Rousseeuw, P. (1990). *Finding groups in data: an introduction to cluster analysis*. New York: John Wiley and Sons.

Mirkin, B. (2005). *Clustering for data mining: a data recovery approach*. Boca Raton, FL: Chapman & Hall/CRC.

ACKNOWLEDGMENTS

I would like to thank Professor Buttrey for allowing me to work with him on this project and for his patience with me throughout this entire process. I would also like to thank Professor Whitaker for her guidance and constant enthusiasm. Finally, I would like to thank my cohort and fellow Operations Research students for getting me through this program. RJ, Trish, Alok, Brennan, Hamadi, Cyrus, Dave, and especially Brett, Jeff, and Cat, I could not have done this without you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THESIS PURPOSE

This thesis is a continuation of work originally presented in Buttrey (2006) titled "A Scale-Independent Clustering Method with Automatic Variable Selection Based on Trees." The purpose of this thesis is to provide three new distance metrics to be used in clustering algorithms that are not influenced by linear transformations, and can be used on datasets with both categorical and continuous variables. Using classification and regressions trees to obtain the distances, our method also allows for automatic variable selection and resistance to noise variables. The data used in this thesis come from University of California at Irvine Machine Learning Repository (Bache & Lichman, 2014).

B. THESIS OBJECTIVES

- Implement an algorithm that calculates three new distance metrics by building classification or regression trees on every variable in the respective datasets.
- Choose an existing clustering algorithm based on the three distance metrics, and use Cramér's *V* to evaluate the quality of the clustering solution.
- Compare the results to other clustering algorithms to evaluate the accuracy of clustering using the new distance metrics.

C. USES OF CLUSTERING

Clustering analysis is the study of determining appropriate groups, or clusters, for a given dataset (Hartigan, 1975, p. 1). Once a distance, or dissimilarity, metric has been established to evaluate the extent of the difference between two observations, clusters of observations are formed. Ideally, observations in a cluster should be very similar to one another and very different from observations in other clusters (Mirkin, 2005, pp. ix–x).

Clustering has been used in various fields including natural science, psychology, and more recently, economics. People have been trying to group different species since Aristotle's time, and taxonomy was finally standardized by Carolus Linnaeus in the 18th century, when he developed the modern system of biological classification (Hartigan, 1975, pp. 1–2). Clustering has also been used in military analysis. Bird and Fairweather (2007) used clustering of casualties of Coalition Forces in Iraq and Afghanistan as a means of predicting future casualties in Afghanistan. Jones et al. (2002) looked at medical records of almost 2,000 British veterans to gather demographic data such as documented medical symptoms and wars in which the veterans fought. They then clustered the data and were able to identify three distinct post-combat syndromes associated with different eras, granting some legitimacy to proposed medical conditions such as the Gulf War Syndrome (Jones et al., 2002, pp. 321–324)

D. MEASURING DISTANCES BETWEEN OBSERVATIONS

Kaufman and Rousseeuw describe a standard clustering scenario using a dataset that contains *n* observations, each with *p* measurements. The measurements can be either continuous or categorical. The *i*th observation of the *k*th measurement is denoted by x_{ik} where $i=1, 2,\dots, n$ and $k=1, 2, \dots, p$. The distance between any two observations *i* and *j* for a given measurement *k* is denoted by $d_k(i,j)$. This distance is used to quantify the dissimilarity between the two observations (Kaufman and Rousseeuw, 1990, pp. 3–4).

1. Euclidean Distance

The most common method to measure dissimilarity between two observations is to calculate the Euclidean distance (Equation 1.1), which is the true geometric distance between observations (Kaufman and Rousseeuw, 1990, p. 11).

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\dots+(x_{ip}-x_{jp})^2} \tag{1.1}$$

The Euclidean distance can be represented by a straight line between the two observations, as illustrated for the two dimensional case in Figure 1 (Hartigan, 1975, p. 58).

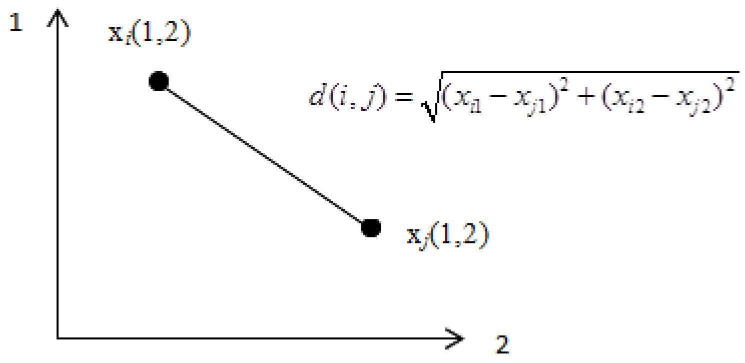

Figure 1. Euclidean distance of two observations (from Kaufmann & Rousseeuw, 1990, p. 12).

2. Manhattan Distance

Another common distance measurement used in clustering is the Manhattan distance, calculated by adding the absolute values of distances between observations for each respective measurement (Equation 1.2).

$$d(i,j)=\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|+\ldots+\left|x_{ip}-x_{jp}\right| \tag{1.2}$$

The distance derives its name from the city streets of Manhattan. Each measurement difference between observations can be represented as a city block in Manhattan, as illustrated in Figure 2 (Kaufman & Rousseeuw, 1990, p. 12).

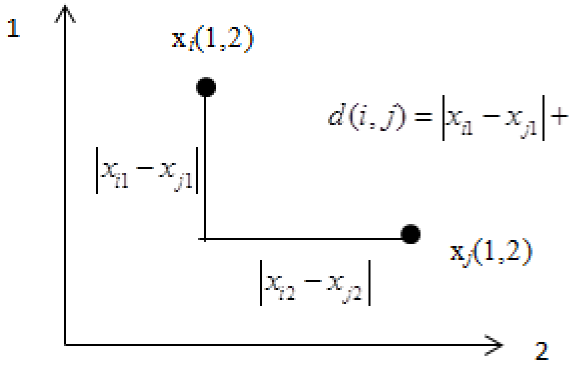
1
$x_i(1,2)$
$d(i,j)=\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|$
$\left|x_{i1}-x_{j1}\right|$
$x_j(1,2)$
$\left|x_{i2}-x_{j2}\right|$
2

Figure 2. Manhattan distance of two observations (from Kaufman & Rousseeuw, 1990, p. 12).

3. Other Distance Metrics

Although most clustering problems use Euclidean or Manhattan distances, many other distance metrics are used to calculate differences between observations in clustering, each used for varying circumstances. Equation 1.3 shows the Minkowski distance, a variation of the Euclidean and Manhattan distances, with *q* taking a value between zero and one (Kaufmann & Rousseeuw, 1990, p. 13).

$$d(i,j) = \left(\left| x_{i1} - x_{j1} \right|^q + \left| x_{i2} - x_{j2} \right|^q + \ldots + \left| x_{ip} - x_{jp} \right|^q \right)^{\frac{1}{q}} \tag{1.3}$$

The distance metrics described above are primarily useful for continuous variables. If a variable is categorical, then dissimilarity is often calculated based solely on whether two observations have the same value for a given measurement, as shown in Equation 1.4 (Hartigan, 1975, p. 64).

$$\begin{aligned} d_k(i,j) &= 0 \quad \text{if } x_{ik} = x_{jk} \\ d_k(i,j) &= 1 \quad \text{if } x_{ik} \neq x_{jk} \end{aligned} \tag{1.4}$$

E. PROBLEMS WITH CURRENT DISTANCE METRICS

For continuous variables, the distance between observations can be influenced by contributing factors such as the units used for each variable and any monotonic transformations made on the data (Kaufmann & Rousseeuw, 1990, p. 4). For both continuous and categorical variables, these distance metrics treat each variable equally, even though some variables might be more important to the dataset than others, and therefore should have a weight applied to them. Variables in the datasets could be correlated, resulting in that attribute contributing to overall distance more than once (Hartigan, 1975, pp. 60–65). Finally, measuring distance between observations when the data has a combination of continuous and categorical variables requires some type of scaling or standardization of the variables (Mirkin, 2005, pp. 65–66).

1. Scaling Measurements

The scale in which a variable is measured is influential when clustering the data (Kaufmann & Rousseeuw, 1990, p. 4). Suppose that a group of ships were clustered based on physical characteristic, with just their displacement and maximum speed used as measurements. Using pounds instead of tons will increase the dissimilarity between ships by increasing the spread of the clusters. One way to avoid this problem is to standardize all of the variables, where each measurement is divided by some standardization value. To avoid the influence of outlying measurements, a standardization method such as the mean absolute deviation (Equation 1.5) is recommended, where m_k is the mean of variable k (Kaufmann & Rousseeuw, 1990, p. 8).

$$s_k = \frac{1}{n}\left(\left|x_{1k} - m_k\right| + \left|x_{2k} - m_k\right| + ... + \left|x_{pk} - m_k\right|\right) \tag{1.5}$$

2. Weighing Variables

One problem with standardizing variables is that it gives each variable the same effect, and therefore, the assumption needs to be made that each variable is equally important in determining clusters in the dataset. Once all of the variables are standardized, weights can be applied to each variable based on importance (Mirkin, 2005, p. 65–66). Using the previous example of the ships, additional measurements including freeboard and height can be used to cluster the ships. If, for example, speed and displacement were better indicators of ship clusters, weights could be assigned to all variables, with heavier weight given to speed and displacement and lighter weight given to freeboard and height. This would result in speed, rather than height, having a greater influence on the dissimilarity between two ships and ultimately the clusters in which they belong. The problem with incorporating weights is that their values can often be subjective. In many cases, a subject matter expert might have to be consulted to determine appropriate weights (Hartigan, 1975, p. 60).

3. Categorical Variables

Determining the dissimilarity between observations using categorical variables is approached in several different ways. First, there is the approach in Equation 1.4; two observations have a distance of 0 if they have the same value for a given variable and a distance of 1 if they have a different value for that variable (Hartigan, 1975, p. 64). Another approach is to convert the entries for a categorical variable into nominal or binary scale. These variables will have to be rescaled due to some entries having more weight than others (Mirkin, 2005, p. 64). For example, if a variable *k* is coded into 1s and 0s, and 0s are much more prevalent, the presence of a 1 might be more significant than the presence of a 0. Observations which have a 1 for *k* might be much more similar to each other than two observations which have a 0 for *k*. Otherwise, it must be assumed that the variable is symmetric and each appearance of 0 and 1 has equal weight. (Kaufmann & Rousseeuw, 1990, p. 26). An appropriate way to scale *k* is to determine the distribution of 1s and use the respective variance for that distribution (Mirkin, 2005, p. 70).

4. Mixed Variables

Some datasets are made of a combination of continuous and categorical variables, which can be difficult to cluster. Kaufmann and Rousseeuw (1990) suggest that a simple solution would be to preform clustering on each of these variables separately and compare the output clusters. This approach is not ideal because different variables could yield different clusters and it might be difficult to determine which clusters are the most accurate. Another method is to approach each variable as if it is on a nominal scale, but then it would have to be assumed that it is symmetric. Finally, as shown in Equation 1.6, each variable could be treated as binary and dissimilarity could be measured using a variation of Equation 1.4 (Kaufmann & Rousseeuw, 1990, p. 34).

$$\begin{aligned} d_k(i,j) &= 0 \quad \text{if } x_{ik} < a_k \\ d_k(i,j) &= 1 \quad \text{if } x_{ik} \geq a_k \end{aligned} \tag{1.6}$$

This method, along with other combinations, can lead to a loss of information. Determining a_k could result in arbitrary clusters because two observations that were not very similar could then be forced into a group together (Kaufmann & Rousseeuw, 1990, p. 34).

F. ADDRESSING CURRENT PROBLEMS

Our clustering method addresses some of the issues with current distance metrics. Our method is not affected by linear transformations, so scaling variables or integrating weights does not affect distance between two observations. We use trees to determine distances between observations, and as described by Faraway (2006) trees are unaffected by linear transformations. Since we scale the resulting deviances anyway, our method is totally unaffected by linear transformations. Trees, furthermore, are unaffected when the predictor variables are transformed in a non-linear but strictly monotonic way. However, a tree built using a non-linearly transformed response will be different than one built with the original response. In this way our method is immune to linear transformation of the input variables and, we might say, "resistant" to monotonic transformation. If a variable is transformed monotonically, trees in which that variable appears as a predictor will be unchanged; the one tree in which it appears as a response will change somewhat (Faraway, 2006, pp. 251–252). Some trees are discarded in our approach (see Chapter II); in this way our method performs automatic variable selection. This way, the analyst is freed from having to consider transformation or variable selection explicitly. Our method does not, however, adjust for correlated input variables.

G. ORGANIZATION OF THE STUDY

Chapter II is a literature review of the tools used in this new clustering method, such as classification and regression trees. Chapter III is an introduction to the new method, a description and examples of the three new distance metrics, and a description of the datasets that were used in the implementation of the new clustering method. Chapter IV goes over the results of our method and compares them to the other clustering algorithms. Chapter V discusses summary and future work to be done on this new clustering method.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

A. CLASSIFICATION AND REGRESSION TREES

The new clustering method uses distances based on results from classification and regression trees (c&rt). According to Hastie, Tibshirani, and Friedman (2001), in a tree, all *n* observations start in the root node. One variable is then chosen for the first split, breaking up into two regions at some split point. Both the variable which is being split and the point at which it is split are chosen to give the tree a best fit. Splits continue to occur on different variables at different split points until the observations are divided into M regions R_1, R_2,…,R_M and a pre-determined stopping point is reached, determined, for example, by a maximum number of nodes (Hastie, Tibshirani, Friedman, 2001, pp. 267–269). While there are other methods for building trees, this thesis only investigates c&rt.

Ooi (2002) addresses the idea of using classification and regression to cluster observations. He proposes an algorithm that builds and prunes trees, as does our method, but he determines his clusters by finding modes and density estimates of the observations in the trees (Ooi, 2002, pp. 328–347).

1. Regression Trees

Regression trees are used for continuous responses. When a response, y, is split on a continuous variable, X_k, it is divided into two regions, R_1: $\{X_k \le c\}$ and R_2: $\{X_k > c\}$, where c is some constant. For X_k, c is chosen so as to minimize the sum of squared errors of the response across the two regions (Equation 2.1). The average y for the left and right child nodes are denoted by $\bar{y}_L$ and $\bar{y}_R$ respectively (Hastie et al., 2001, p. 269).

$$D = \sum_{i:X_{ik} \le c} (y_i - \bar{y}_L)^2 + \sum_{i:X_{ik} > c} (y_i - \bar{y}_R)^2 \qquad (2.1)$$

Every possible split on variable k is considered, and the process is repeated for all continuous predictors. The split produces two subsequent nodes, and splitting is considered on every node, using the same criterion, until the stopping point is reached.

For categorical predictors, a split produces two nodes which are determined by considering every possible binary split for all levels of that variable, ultimately choosing the one which produces the greatest change in deviance. The stopping point may not yield an optimal tree, however, either stopping with a tree that is too small or one that is too large and has over fit the data. In general, the tree is overgrown and must be pruned to reach optimality, and therefore a pruning method must be established (Hastie et al., 2001, p. 269).

2. Classification Trees

Hastie et al. (2001) approach classification trees differently than regression trees, although the former are built using a similar algorithm. A split on a node for response *y* should still in result reducing the deviance of the response across the two resulting regions. Instead of using the regression tree criterion of Equation 2.1, classification trees use deviance, also called cross-entropy. For a response with *j* classes, the deviance is defined by Equation 2.2.

$$D = -2\sum_{j=1}^{J} n_j \log(\hat{p}_j) \tag{2.2}$$

This criterion looks at the proportion of class *j* for the response variable *y* in node R_m. The number of observations of class *j* is denoted by n_j and the estimated proportion of class *j* is denoted by $\hat{p}_j$, which can be calculated by $\hat{p}_j = \frac{n_j}{n}$ (Hastie et al., 2001, p. 271). For this summation, we take $0 * \log(0) = 0$.

B. MEASURING QUALITY OF TREES

Trees can often become larger than necessary, over-fitting the data. When this over-fitting occurs, the trees need to be pruned to an optimal size. Our method uses the K-fold cross-validation rule to determine an appropriate tree size.

1. K-fold Cross-Validation

The K-fold cross-validation method is used to estimate the prediction error of a method by estimating the extra-sample error. The extra-sample error, Err, is the generalized error found when method $\hat{f}(\ )$ is applied to a test set taken from the original data set (Hastie et al., 2001, p. 214). For a general loss function *L*( ), we have Equation 2.3, where the expectation is taken over the joint distribution of the response and predictors (Y, X).

$$Err = \mathrm{E}\left[L(Y, \hat{f}(X)) \right] \tag{2.3}$$

K-fold cross-validation gets its name by dividing the data into *k* equal sections. One of the *k* sections is set aside as a test set and the other $k - 1$ sections are used as a training set. The model is then fit to the remaining sections in the training set. Next, the test set is predicted and the prediction error of the training set is calculated (Hastie et al., 2001, p214). The fitted model of the training set is denoted by $\hat{f}(\ )$. Finally, the cross-validation estimate of prediction error is calculated for a continuous predictor using Equation 2.4 (Faraway, 2006, p. 213).

$$CV = \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 \tag{2.4}$$

For a categorical predictor, each item in the test set is assigned a class using the tree built using the training set; then Equation 2.2 is applied to compute the deviance.

For our clustering method, we use 10-fold cross validation, leaving 10 percent of the data aside to be used a test set each time. The datasets that are used in the model implementation are large enough that the cross-validation should not be biased by the removal of the test set. We prune each tree to the size that minimizes the cross-validated prediction error. If a tree is pruned to the root—that is, if cross-validation indicates that no split is predictive of the response—then that tree is omitted. A variable whose tree is discarded, and which never appears as a predictor in any other tree, is omitted. In this way our algorithm performs variable selection automatically.

2. One-Standard Error Rule

Although we do not find it necessary to use in our new method, another way to prune trees is the one-Standard Error (one-SE) rule. According to Hastie et al (2001), the cross-validation method produces standard errors of the misclassification error rates for the *k* sections. The one-SE rule is often applied to decide the best model from the cross-validation. The criteria for selection is that it has to be the best model out of all *k* models which have errors within one SE of the model with the smallest error (Hastie et al., 2001, p. 215).

C. CLUSTER EVALUATION

There needs to be a way to establish whether our clustering method accurately groups observations into their true classes. For this reason, we use well-known datasets for which the true classes of each observation are known and evaluate how well our method is able to cluster the observations into their respective classes. One method of evaluating the prediction level, how accurate our method is at clustering the data, is a Pearson's chi-squared test goodness of fit statistic, χ^2 (Faraway, 2006, p. 40). We use a normalization of Pearson's chi-squared statistic, Cramér's *V* (Equation 2.5), as a goodness-of-fit statistic for our method. The number of true classes is represented by *K* while *C* is the number of classes produced by the clustering algorithm, and *n* is the number of observations.

$$V = \frac{\chi^2}{n * \min(K-1, C-1)} \tag{2.5}$$

III. METHODOLOGY

A. INTRODUCTION

The major concept of our clustering method is that dissimilarity between two observations can be determined by how often they fall into the same leaf of a classification or regression tree. Observations are similar if they often fall in the same leaf and are different if they usually fall in two different leaves.

B. MODEL ASSUMPTIONS

Our method builds classification and regression trees for each variable in the dataset. Ten-fold cross-validation is used to prune the tree to ensure that a tree with a reasonably small error is created. This tree is taken to be the optimal-sized tree for that variable.

Once the distances between all pairs of observations are calculated, a dissimilarity matrix is constructed. A final clustering algorithm based on these distances is implemented. For our method, we use agglomerative nesting (AGNES) (Kaufman and Rousseeuw, 1990).

C. METHOD IMPLEMENTATION

Clustering algorithms do not use any specific variable as a response variable, so our method starts by creating p trees, one with each variable used as the response variable. The trees are pruned using 10-fold cross-validation. Every observation is assigned to one of the resulting leaves in each tree. The assignment of leaf l for tree t is used to determine the distance between two observations, based on one of the three new distance metrics. All three distance metrics operate under the notion that observations are similar to other observations on the same leaf and different to those observations on other leaves.

1. Distance Metric 1, d_1

The first of the new distance metrics, d_1, is relatively straightforward (Equation 3.1). For tree t, two observations have a distance of 1 if they fall on different leaves. If the observations fall on the same leaf, they have a distance of 0. The leaf on which observation i falls for tree t is denoted by $L_t(i)$.

$$d_1(i,j) = \sum_{t=1}^{p} \left\{ \begin{matrix} 0 & \text{if } L_t(i) = L_t(j) \\ 1 & \text{if } L_t(i) \neq L_t(j) \end{matrix} \right\} \tag{3.1}$$

After all the p trees are constructed, the distance between two observations is the number of trees in which the observations fall on different leaves.

2. Distance Metric 2, d_2

The second distance metric, d_2, applies the same idea as the first: observations on the same leaf are similar to each other and different from observations on other leaves. This simplified idea, however, treats all trees equally. Although all trees are pruned, some trees might actually be better than others. The measure of the quality of a tree is the overall decrease in deviance that it produces, based on the difference between the deviance at the root node, D_t, and the sum of the deviances of all the leaves of tree t. The difference in deviance is denoted by ΔD_t.

A tree with a large ΔD_t is assumed to be of better quality than a tree with a smaller ΔD_t. Therefore, if two observations are on different leaves of a good quality tree, they are perhaps more dissimilar than two trees that land on different leaves of a poor quality tree. Once all p trees are constructed, we determine which tree has the greatest ΔD_t, $\max_t(\Delta D_t)$, and establish that this is the strongest tree of the dataset. The changes in deviance for the remaining trees are scaled by $\max_t(\Delta D_t)$. Equation 3.2 shows the formulation for d_2.

$$d_2(i, j) = \sum_{t=1}^{p} \begin{Bmatrix} 0 & \text{if } L_t(i) = L_t(j) \\ \dfrac{\Delta D_t}{\max\limits_t(\Delta D_t)} & \text{if } L_t(i) \neq L_t(j) \end{Bmatrix} \qquad (3.2)$$

Figure 3 shows an example of one of p trees that might be used to calculate dissimilarities using d_2. The deviance of each leaf is shown in the large ovals and the leaf number in small circles. For this example, assume that another tree yields a the greatest change in deviance for all p trees and $\max\limits_t(\Delta D_t) =$ 12000. If observation i and observation j both land in the same leaf in this tree, they have a distance of 0, just as with d_1. However, if they fall in different leaves, their distance from each other is (10000-4700)/12000 = 0.44. The sum of the deviance of all of the leaves is 4700 and D_t = 10000 for this tree. Scaling all of the changes in deviance by $\max\limits_t(\Delta D_t)$ allows the best tree to have the most weight in determining distance between observations. If two observations fall in different leaves in that tree, we assume that their difference is better represented in that strong tree than in a tree that is not as good. For this example, the distance between two observations that were in different leaves in the best tree would be 12000/12000 = 1 for that tree. This is the maximum distance between two observations on any tree.

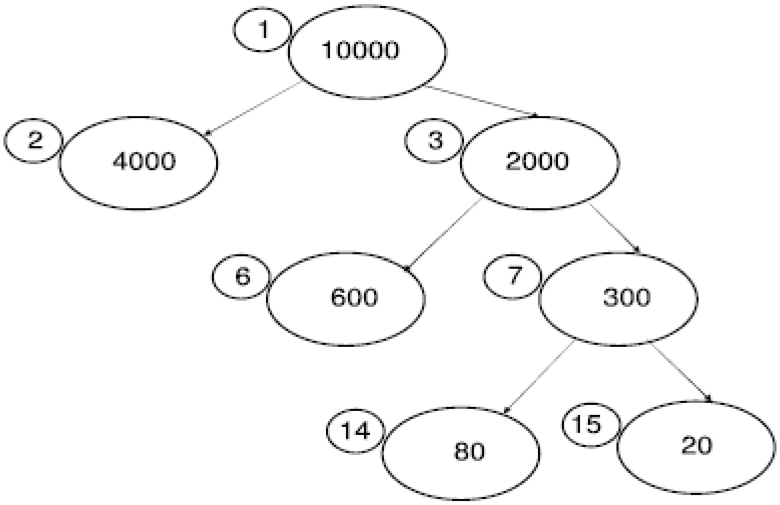

Figure 3. Tree with deviance in large circles and leaf number in small circles (from Buttrey, 2006)

3. Distance Metric 3, d_3

The third distance metric, d_3, calculates distance based on deviance of leaves as in d_2, but unlike the first two distance metrics, does not assume that all leaves are equally different from each other. Instead, d_3 (Equation 3.3) looks at the change in deviances of the tree only up to the shared parent node of the two leaves whose distance is being evaluated, in addition to the change in deviances of the whole tree, ΔD_t, as before. The change in deviances of the partial tree up to the parent node is denoted by $\Delta D_t(i, j)$.

$$d_3(i, j) = \sum_{t=1}^{p} \left\{ \begin{array}{ll} 0 & \text{if } L_t(i) = L_t(j) \\ \dfrac{\Delta D_t(i, j)}{\Delta D_t} & \text{if } L_t(i) \neq L_t(j) \end{array} \right\} \tag{3.3}$$

As with the first two distance metrics, observations *i* and *j* have a distance of 0 if they fall on the same leaf. With d_3, observations which fall on leaves that are separated by multiple splits are deemed farther apart. Figure 4 shows an example of a tree where the distance between leaf 14 and leaf 15 is evaluated under d_3. The total change in deviance for the entire tree is the maximum deviance of the tree, 10000, minus the sum of the deviances of the leaves, 4700, or $\Delta D_t = 10000 - 4700 = 5300$. Next, the tree is cropped at the parent node of 14 and 15, leaf 7. Now, $\Delta D_t(i, j)$ is calculated by subtracting the original sum of deviances of all of the leaves, 4700, from the sum of deviances of the newly cropped tree, 4900, or $\Delta D_t(i, j) = 4900 - 4700 = 200$. The total distance between leaf 14 and leaf 15 is $\dfrac{\Delta D_t(i, j)}{\Delta D_t} = \dfrac{200}{5300} = 0.038$.

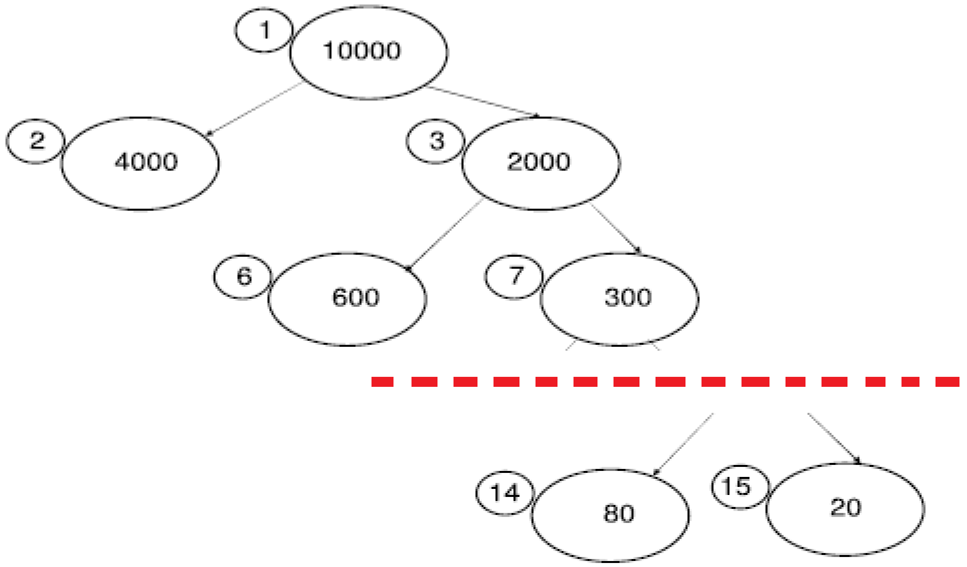

Figure 4. Tree with distance between leaf 14 and leaf 15 evaluated using d_3.

Observations which fall in leaf 14 and leaf 15 probably do share many similar qualities, as they are only separated by one split. Following this logic, observations in leaf 2 and leaf 14 are probably as dissimilar as possible since they are removed from each other by the maximum number of splits. These leaves in fact have the maximum distance of 1, $\frac{\Delta D_t(i, j)}{\Delta D_t} = \frac{10000 - 4700}{10000 - 4700}$ =1. The shared parent node for leaf 2 and leaf 14 is the root node, which has the maximum deviance for this tree.

D. CLUSTERING ALGORITHMS

1. Final Clustering Method AGNES

Once all of the distances between observations are calculated using one of the new three distance metrics, we are left with a dissimilarity matrix of all of the pair-wise distances. A final clustering algorithm based on these distances produces the final clusters.

Kaufman and Rousseeuw (1990) describe an agglomerative method of clustering, AGNES, in which there are originally *n* clusters, each with one observation. These clusters are then successively merged together until there is one cluster which contains all *n* observations. AGNES can be used for a set of interval-scaled variables or a dissimilarity matrix, as used in our method. AGNES specifically looks at distances between clusters, which is why each observation starts as its own cluster. AGNES works in a sequence of steps where the two closest clusters are joined and now treated as one cluster. Then the next two closest clusters are joined and so on until all *n* clusters form one cluster (Kaufman & Rousseeuw, 1990, pp. 202–205).

Although AGNES combines *n* clusters into one cluster, it can be given a threshold by which to create *k* clusters. The datasets on which our new method is tested have the observations labeled into *k* clusters according to classification. Although these classes are pre-determined, there might be more clusters in nature than there are classes (Kaufman and Rousseeuw, 1990, p. 199), so our method is tested by clustering the data into *k* and 2*k* clusters.

2. Other Clustering Algorithms

We also run four well-known clustering algorithms on the datasets, both using *k* and 2*k* clusters. Partitioning around medoids (PAM), divisive analysis (DIANA), and AGNES were applied to all three datasets. *K*-means can only be used on numerical data and therefore was not applied to the Splice dataset. See Kaufmann and Rousseeuw (1990) for more information on PAM and DIANA, and Hartigan (1975) for more information on K-means.

E. DATASETS

This thesis tests the new clustering method on three well-known datasets, Table 1, from the University of California at Irvine Machine Learning Repository (Bache & Lichman, 2014). The true classification of the observations is known for each of these datasets, so this allows us to test our method's predictive power. For each dataset, 15 and 50 variables of random noise were added to test the new method's resilience to noise. Each dataset and the generation of the noise variables are explained in detail in the following sections.

Name	Observations	Variables	Data Type	Classes
Iris	150	4	Numeric	3
Iris with noise	150	19	Numeric	3
Optical	1797	64	Numeric	10
Optical with noise	1797	79	Numeric	10
Splice	3190	60	Categorical	3
Splice with noise	3190	75	Categorical	3

Table 1. Dimensions of datasets used for validation.

1. Iris

The Iris dataset originally comes from Sir Ronald Fisher's Iris Plants Database. It has 150 observations of different irises, with measurements of sepal length, sepal width, petal length, and petal length, all measured in centimeters. The observations are classified into three classes of species, with 50 observations per species. This dataset has been used in numerous publications and is one of the most well-known datasets for classification and pattern recognition (Bache & Lichman, 2014).

The variables of random noise were generated in R using a random generation from the normal distribution. Each noise variable has a length of 150 to correspond with the observations of the dataset. The mean of the distribution is 0 and the standard deviation is 1.

2. Optical

The Optical dataset is a test set from Ethem Alpaydin and Cenk Kaynak's dataset titled "Optical Recognition of Handwritten Digits." The original dataset has 5620 observations; we use approximately one-third of these observations for testing purposes. The test set gives a good representation of the different clusters without being too computationally taxing. This dataset uses handwritten digits zero through nine from 13 different people. The digits are converted to a digitized form and then preprocessing programs are used to try to optically recognize the digits (Bache & Lichman, 2014).

Although the hand-written digits are zero through nine, the inputs for recognition, after being converted to a digitized form, were integers zero through 16 (Bache &

Lichman, 2014). The noise variables were generated in R by drawing a random sample, with replacement, of integers zero through 16. Each noise variable has a length of 1797.

3. Splice

The final dataset, Splice, comes from the Genbank 64.1 database "Primate Splice-Junction Gene Sequences (DNA) with Associated Imperfect Domain Theory." The dataset is made of 3190 observations of DNA sequences, each with 60 DNA sequence elements. The observations are divided into three classes of splice sites, intron-extron, extron-intron, or neither. These classes represent the point in the DNA sequence in which unnecessary DNA sequence elements are removed (Bache & Lichman, 2014).

The noise variables added to Splice were generated in R by drawing a random sample with replacement of the letters C, A, G, and T to correspond to the DNA sequence elements. Each noise variable has a length 3190 to fit with the rest of the dataset.

IV. ANALYSIS

A. INTRODUCTION

This chapter presents the results of our new clustering method for all three distance metrics, using Cramér's *V* to measure how well our method accurately groups each observation into their true classes. We compared these results to the Cramér's *V* values from running AGNES, PAM, DIANA, and *K*-means on the datasets.

B. RESULTS

The clustering algorithms were run on a mid-grade laptop with a dual core processor, a 64-bit operating system, and four gigabytes of RAM. Each run used 64-bit R, Version 2.15.1 (R Core Team, 2012). For each dataset, the existing algorithms—AGNES, DIANA, PAM, and *K*-means—took five minutes or less to run. That was also the case with our new measures d_1 and d_2. In the Iris data set, the d_3 measure took approximately the same time as the other algorithms (that is, almost no time). However, d_3 tended to be quite a bit slower than the existing algorithms and slower than d_1 and d_2. The d_3 algorithm took less than 10 minutes on the Optical and Splice datasets with no noise (so approximately double the time required for the existing algorithms), about 20 minutes for the datasets with 15 noise variables, and approximately 40 minutes when 50 noise variables were added to the Optical dataset. The d_3 algorithm could not be run on the laptop with the Splice data plus 50 noise variables, due to insufficient memory. Instead, a high-powered workstation (192 gigabytes of RAM) was necessary, and on that platform, approximately 15 minutes was required. Of course, since our algorithms operate independently on columns, they are well-suited to parallel processing.

Table 2 shows the Cramér's *V* of all of the clustering algorithms on the Iris, Optical, and Splice datasets, with different amounts of noise and both with *k* and 2*k* clusters. *K*-means cannot cluster numerical data, so no results are shown for this algorithm with the Splice data. The highest values in each row are highlighted.

Dataset	Clusters	Agnes	Diana	Pam	Kmeans	d1	d2	d3
Iris	3	0.781	0.709	0.745	0.745	0.781	0.769	0.649
Iris	6	0.785	0.798	0.858	0.785	0.878	0.877	0.876
Iris 15 noise variables	3	0.464	0.5	0.444	0.541	0.649	0.877	0.649
Iris 15 noise variables	6	0.483	0.535	0.471	0.459	0.92	0.896	0.877
Iris 50 noise variables	3	0.02	0.5	0.22	0.486	0.521	0.685	0.649
Iris 50 noise variables	6	0.045	0.482	0.3	0.505	0.876	0.896	0.65
Optical	10	0.505	0.472	0.659	0.607	0.673	0.52	0.477
Optical	20	0.728	0.707	0.769	0.831	0.748	0.727	0.711
Optical 15 noise variables	10	0.426	0.454	0.575	0.568	0.574	0.565	0.587
Optical 15 noise variables	20	0.724	0.664	0.742	0.796	0.7	0.718	0.719
Optical 50 noise variables	10	0.495	0.529	0.498	0.552	0.56	0.582	0.479794
Optical 50 noise variables	20	0.709	0.673	0.592	0.81	0.75	0.713	0.692617
Splice	3	0.0004	0.0751	0.0582		0.0017	0.3571	0.0016
Splice	6	0.0016	0.1077	0.0579		0.3834	0.6786	0.3387
Splice 15 noise variables	3	0.001	0.0292	0.0486		0.0017	0.3571	0.0016
Splice 15 noise variables	6	0.0023	0.1326	0.062		0.3834	0.6786	0.3387
Splice 50 noise variables	3	0.001	0.0916	0.0422		0.0017	0.3571	0.0394
Splice 50 noise variables	6	0.004	0.1326	0.0513		0.3834	0.6786	0.5811

Table 2. Cramér's *V* for the different clustering algorithms for *k* and 2*k* clusters with highest values high-lighted.

Out of the 18 scenarios, at least one of the new distance metrics performed better than the other clustering algorithms in 77 percent of the cases. With the exception of Iris with no noise and three clusters, each new distance metric did better than all four other clustering algorithms for the Iris dataset. In general, the other algorithms performed far worse with the introduction of the 50 noise variables. In those cases, d_2 performed better than the other methods.

For the Optical dataset, *K*-means performed better than all other algorithms when using 20 clusters. With 10 clusters, each new distance metric did the best, depending on the amount of noise added to the dataset. When no noise was added, d_1 performed the best, d_3 performed the best when 15 noise variables were added to the data, and once again when 50 noise variables were added, d_2 performed better than the other distance metrics.

For every case in the Splice data, our second distance metric, d_2, did better than the other algorithms and the other two new distance metrics. The other two distance metrics, d_1 and d_3, did better than the other four algorithms when using six clusters. The difference in performance was minimal when the data was grouped into three clusters.

C. CONCLUSION

Our new clustering method performed better than current clustering algorithms in a majority of cases. For the Iris and Splice datasets, the distance metrics performed much better than the other algorithms in the presence of noise. When noise was added to the Optical dataset, d_2 and d_3 performed better than the other methods when clustering with 10 clusters. *K*-means, however, performed better in the presence of noise with 20 clusters. We had expected that our method would do better with large amounts of noise due to the variable selection that automatically occurs.

THIS PAGE INENTIONALLY LEFT BLANK

V. SUMMARY AND FUTURE WORK

A. SUMMARY

The purpose of this thesis was to provide a new clustering method that was not influenced by linear transformations and was able to perform automatic variable selection. Chapter I introduced clustering and some commonly used distance metrics and described some problems with current clustering methods. Chapter II focused on tools that we used in our new method. Chapter III described our method, gave examples of each of the new distance metrics, and described the datasets which were used to test our clustering method.

Our new method performs very well at accurately clustering datasets into their true classes, especially in the presence of noise. In 18 different scenarios, our method performed better than four other clustering algorithms over 77 percent of the time. The use of classification and regression trees eliminates the need to scale the variables and allows for easy clustering of data with mixed variables. The only scaling that occurs does so automatically when using d_2 or d_3 when each distance measurement is scaled by the change in deviance of the best tree. Variables are automatically selected based on the quality of their respective trees. Automatic variable selection is one reason that our method out-performs other clustering algorithms when noise variables are present.

B. FUTURE WORK

Each distance metric is more sophisticated than the last and we anticipated that this increase in complexity would yield more accurate distances between observations. We predicted that d_2 would perform better than d_1 but expected that d_3 would do the best out of the three distance metrics. This was only the case when the Optical dataset with 15 noise variables was clustered into 10 clusters. The difference between d_3 and the other two distance metrics is that it takes into account that observations which fall in leaves separated by a small number of splits are more similar than observations which fall in leaves separated by a large number of splits. Future work remains to be done on this idea to see if it actually should yield more accurate results than d_1 and d_2.

Breiman (1996) introduced bootstrap aggregation (bagging), a method which produces multiple outcomes of a predictor and averages over the outcomes for numerical values and uses a plurality vote for categorical data. Breiman used bagging on classification and regression trees as a means to reduce misclassification error (Breiman, 1996, pp. 123–125). More work is to be done on our new method to apply bagging into the algorithm. Instead of producing one tree per variable, we believe that producing multiple trees per variable and averaging the distances of those trees might produce a more accurate representation of distances between observations. Parallelization would work very well when using bagging. Using p computers for p variables would cut down on computational time. In fact, parallelization should bring substantial speed benefits, even in our existing algorithm.

Finally, our new method uses AGNES for a final round of clustering. AGNES might not be the best choice as a final clustering algorithm, especially for large datasets. Work is still to be done on determining the most optimal final clustering algorithm.

LIST OF REFERENCES

Bache, K. & Lichman, M. (2014). *UCI Machine Learning Repository*. Retrieved from University of California at Irvine, School of Information and Computer Science website: http://archive.ics.uci.edu/ml/datasets.html

Bird, S. M., & Fairweather, C. B. (2007). Military fatality rates (by cause) in Afghanistan and Iraq: A measure of hostilities. *International Journal of Epidemiology, 36*(4), 841–846.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Buttrey, S. (2006). *A scale-independent clustering method with automatic variable selection based on trees*. Unpublished manuscript, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

Faraway, J. (2006). *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models.* Boca Raton, FL: Chapman & Hall/CRC.

Hartigan, J. (1975). *Clustering algorithms*. New York: John Wiley and Sons.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* New York: Springer-Verlag.

Kaufman, L. & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: John Wiley and Sons.

Jones, E., Hodgins-Vermaas, R., McCartney, H., Everitt, B., Beech, C., Poynter, D., Palmer, I., Hyams, K., & Wessely, S. (2002). Post-combat syndromes from the Boer War to the Gulf War: A cluster analysis of their nature and attribution. *BMJ: British Medical Journal, 324*(7333), 321–324.

Mirkin, B. (2005). *Clustering for data mining: a data recovery approach*. Boca Raton, FL: Chapman & Hall/CRC.

Ooi, H. (2002). Density visualization and mode hunting using trees. *Journal of Computational and Graphical Statistics*, 11(2), 328–347.

R Core Team. (2012). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California